

REMARKS

By this Amendment, Applicant amends claim 1 to more appropriately define the invention recited therein. Claims 1-16 remain pending in this application. In the Office Action of July 28, 2005,¹ claims 1, 5, and 13-16 were rejected under the judicially created doctrine of obviousness-type double patenting as unpatentable over claims 1-4 of U.S. Patent No. 6,802,057 B1; claims 1-16 were rejected under 35 U.S.C. § 102(a) as anticipated by the document entitled “64-Bit Application Development for PA-RISC & IA-64” (“*Coutant*”); and claims 13-16 were rejected under 35 U.S.C. § 102(b) as anticipated by the document entitled “*Microsoft* Interface Definition Language (MIDL): 64-Bit Porting Guide” (“*Microsoft*”). Applicant addresses the rejections below.

Obviousness-type double patenting rejection of claims 1, 5, and 13-16

Although disagreeing with the obviousness-type double patenting rejection of claims 1, 5, and 13-16, in an effort to advance prosecution, Applicant files a Terminal Disclaimer concurrently with this paper, obviating the outstanding rejection. Applicant requests reconsideration and withdrawal of the obviousness-type double patenting rejection of these claims in view of the attached Terminal Disclaimer.

Applicant points out that: “[t]he filing of a terminal disclaimer to obviate a rejection based on nonstatutory double patenting is not an admission of the propriety of the rejection.” M.P.E.P. § 804.02(II), 8th Ed., Aug. 2001, p. 800-32 (citing *Quad Environmental Tech. Corp. v. Union Sanitary Dist.*, 946 F.2d 870 (Fed. Cir. 1991)). As M.P.E.P. § 804.02(II) indicates, “[t]he Court indicated that the ‘filing of a terminal disclaimer simply serves the statutory function of

¹ The Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether or not any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

removing the rejection of double patenting, and raises neither a presumption nor estoppel on the merits of the rejection.”” *Id.*

Section 102(a) rejection of claims 1-16

I. Regarding the Coutant document

The Examiner has not established that *Coutant* qualifies as prior art under 35 U.S.C. § 102(a). The *Coutant* document appears to be a printout of an electronic presentation and bears the date of March 17, 2000. To the extent the Examiner is applying *Coutant* as a “printed publication” within the meaning of § 102(a), Applicant reminds the Examiner of the requirement for “a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it.” M.P.E.P. § 2128 (internal citations omitted). Further, an electronic publication cannot be relied upon as prior art under § 102(a) if it does not include a publication date or retrieval date. *See* M.P.E.P. § 2128. In this case, the Examiner has not established that the date of March 17, 2000, is relevant to the *Coutant*’s availability, publication, or retrieval. Unless the Examiner produces the requisite proof of *Coutant*’s dissemination and availability prior to Applicant’s filing date (*See* M.P.E.P. § 2128), *Coutant* is not a competent prior art reference within the context of § 102(a) and cannot be used to reject Applicant’s claims.

II. Regarding the merits of the rejection

Regardless of whether *Coutant* is in fact a proper § 102(a) reference, Applicant traverses the § 102(a) rejection of claims 1-16 because *Coutant* fails to anticipate the claims. In order to properly anticipate Applicant’s claimed invention under § 102, each and every element of the claim at issue must be found, either expressly described or under principles of inherency, in a

single prior art reference. Further, “[t]he identical invention must be shown in as complete detail as is contained in the . . . claim.” *See* M.P.E.P. § 2131. Finally, “[t]he elements must be arranged as required by the claim.” *Id.*

With regard to claim 1, *Coutant* fails to teach at least the following features:

adding to the interface file a directionality of at least one of the integer parameter and the logical parameter based on comments in the source code;

adding to the interface file a parameter size along each dimension of at least one of the integer parameter and the logical parameter;
and

reading the interface file to generate a stub routine that converts at least one of the integer and logical parameters from 32-bit to 64-bit and that invokes the subprogram by specifying the converted parameters.

Coutant describes a 64-bit programming model. It addresses porting to 64-bits, as well preparing code for 64 bits. *Coutant* compares ILP32 (integer, long, and pointer) and the LP64 programming models, noting that in LP64 long and pointer types are 64 bits and also that LP64 includes certain extended derived types (pages 3-4). *Coutant* also compares aspects of 32-bit and 64-bit runtime environments, commenting that with 64-bit code the compiler can inline import stubs (pages 5-6). As the Examiner noted, *Coutant* also mentions accessing a “linkage table” in a program for globals (OA at 4; *Coutant* at 11).

Although *Coutant* compares 32-bit and 64-bit programming models, the document does not teach at least the “adding” and “reading” features of claim 1. Applicant reminds the Examiner that a rejection under § 102 is proper only when the claimed subject matter is identically described or disclosed in the prior art. *In re Arkley*, 455 F.2d 586, 587, 172 USPQ 524, 526 (CCPA 1972). As noted above, “[t]he identical invention must be shown in as complete detail as is contained in the . . . claim.” M.P.E.P. § 2131. In this case, *Coutant* does

not identically describe the subject matter of claim 1 “in as complete detail as is contained in the . . . claim.”

Furthermore, even if all of the features of claim 1 could be found in various teachings of *Coutant* – Applicant disputing that contention – the reference does not clearly and unequivocally disclose the claimed invention or direct those skilled in the art to the claimed invention “without any need for picking, choosing, and combining various disclosures.” *In re Arkley*, 455 F.2d 586, 587,172 USPQ 524, 526 (CCPA 1972) (emphasis added).

For at least the foregoing reasons, *Coutant* does not support the § 102(a) rejection of claim 1, as asserted by the Examiner. As such, the § 102(a) rejection of claim 1 based on *Coutant* should be withdrawn.

With regard to independent claim 3, *Coutant* fails to teach at least “automatically generating a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code,” as asserted by the Examiner. Although *Coutant* compares 32-bit and 64-bit programming models, the reference does not teach at least automatically generating a 32-bit to 64-bit conversion stub. Contrary to the Examiner’s position (OA at 4), *Coutant*’s description of a comparison of 32-bit and 64-bit runtime environments at pages 5 and 6 does not constitute “automatically generating a 32-bit to 64-bit conversion stub,” as claimed. Because *Coutant* does not support the § 102(a) rejection of claim 3, as asserted by the Examiner, the rejection should be withdrawn.

With regard to independent claim 5, *Coutant* fails to teach at least the following features:

an interface generator that reads the subprogram and that generates
an interface file with indications of characteristics of the
parameter; and

a stub generator that reads the interface file and that generates a
stub for the subprogram by using the characteristics, wherein each

of the stubs receives a set of parameter values, generates the values for the required parameters from the received set of parameter values, and invokes the subprogram with the values for the parameters.

In rejecting claim 5, the Examiner noted *Coutant*'s comparison of the ILP32 and LP64 programming models (OA at 5; *Coutant* at 3-4). The Examiner alleged that in *Coutant* "the 'type' is identified as characteristics for the developing of 32-bit code and 64-bit code in conversing/porting" (OA at 5). Applicant disagrees with the Examiner's interpretation of *Coutant*. To begin with, even if *Coutant* were to disclose parameter "characteristics," the document does not teach "an interface generator that reads the subprogram and that generates an interface file with indications of characteristics of the parameter" and "a stub generator that reads the interface file and that generates a stub for the subprogram by using the characteristics . . .," as asserted by the Examiner. Instead, *Coutant* merely compares ILP32 and LP64, noting that in LP64 long and pointer types are 64 bits and also that LP64 includes certain extended derived types (pages 3-4). Further, *Coutant* does not disclose "developing" 32-bit code or converting between 64-bit and 32-bit code, and the Examiner provides no evidence to support the allegation that the identified "types" facilitate such developing and converting. Additionally, *Coutant*'s comparison of the 32-bit and 64-bit runtime environments at pages 5 and 6 does not constitute "an interface generator" and "a stub generator," as recited in claim 5.

For at least the foregoing reasons, *Coutant* does not support the § 102(a) rejection of claim 5. The § 102(a) rejection of claim 5 based on *Coutant* should therefore be withdrawn.

Independent claims 13 and 16, although of different scope than claim 3 (and from each other), include features similar to those of claim 3. In particular, claim 13 recites, *inter alia*, "receiving 32-bit source code" and "automatically generating a 32-bit interface to 64-bit source code." Claim 16 recites, *inter alia*, "means for receiving 32-bit source code" and "means for

automatically generating a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.” For at least reasons similar to those presented above in connection with claim 3, claims 13 and 16 are distinguishable from *Coutant*. Accordingly, the § 102(a) rejection of claims 13 and 16 should be withdrawn.

With regard to claim 15, *Coutant* fails to teach at least “generating a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter,” as asserted by the Examiner. Contrary to the Examiner’s position, *Coutant*’s comparison of the 32-bit and 64-bit runtime environments at pages 5 and 6 does not constitute “generating a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter,” as claimed. Indeed, neither the cited portions nor any other portions of *Coutant* teach the “generating” feature of claim 15. Because *Coutant* does not support the § 102(a) rejection of claim 15, the rejection should be withdrawn.

Claim 2 depends upon claim 1; claim 4 depends upon claim 3; claims 6-12 depend upon claim 5; and claim 14 depends upon claim 13. Claims 2, 4, 6-12, and 14 are distinguishable from *Coutant* for at least reasons similar to those presented above in connection with claims 1, 3, 5, and 13. The § 102(a) rejection of claims 2, 4, 6-12, and 14 should therefore be withdrawn.

Section 102(b) rejection of claims 13-16

I. Regarding the Microsoft document

The Examiner has not established that *Microsoft* qualifies as prior art under 35 U.S.C. § 102(b). The *Microsoft* document appears to be a printout of an Internet file, presumably extracted from a larger work. The document bears a “Last updated” date of August 1999 and refers to a technical white paper, which appears to claim a copyright date of 1998 (see page 18). To the extent the Examiner is applying *Microsoft* as a “printed publication,” Applicant

reminds the Examiner of the requirement for “a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it.”

M.P.E.P. § 2128 (internal citations omitted). Further, an electronic publication cannot be relied upon as prior art under if it does not include a publication date or retrieval date. *See* M.P.E.P. § 2128. The Examiner has not established *Microsoft*’s 1998 and 1999 dates are relevant to its availability, publication, or retrieval. Unless the Examiner produces the requisite proof of *Microsoft*’s dissemination and availability more than one year prior to Applicant’s filing date (*See* M.P.E.P. § 2128), *Microsoft* is not a competent prior art reference within the context of § 102(b) and cannot be used to reject Applicant’s claims.

II. Regarding the merits of the rejection

Applicant traverses the § 102(b) rejection of claims 13-16 for the following reasons.

With regard to claim 13, *Microsoft* fails to teach or suggest at least “automatically generating a 32-bit interface to 64-bit source code,” as claimed. The Examiner notes *Microsoft*’s disclosure regarding various attributes and a “64-Bit Stub Generation Model” (OA at 7; *Microsoft* at 12-15). These cited portions of *Microsoft* do not teach the “automatically generating” feature of claim 13, as asserted by the Examiner. *Microsoft* describes a compiler generating a stub for a given environment (page 15). *Microsoft* also describes a “dual stub,” which consists of 32-bit and 64-bit parts and is “a convenient tool for porting . . .” (page 15). The *Microsoft* document does not teach “automatically generating a 32-bit interface to 64-bit source code,” as alleged in the Office Action. Merely generating a stub for a given environment and using dual stubs for porting does not constitute “automatically generating a 32-bit interface to 64-bit source code,” as claimed.

Because *Microsoft* does not support the § 102(b) rejection of claim 13, the rejection should be withdrawn. Claim 14 depends upon claim 13 and, therefore, is likewise distinguishable from *Microsoft*. Accordingly, the § 102(b) rejection of claim 14 based on *Microsoft* should also be withdrawn.

With regard to independent claim 15, *Microsoft* fails to teach at least “reading the source code” and “generating a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter,” as asserted by the Examiner. Contrary to the Examiner’s position (OA at 7), generating a stub for a given environment and using dual stubs for porting, as disclosed by *Microsoft*, does not constitute “reading the source code” and “generating a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter.” Because *Microsoft* does not support the § 102(b) rejection of claim 15, Applicant requests withdrawal of the rejection.

Independent claim 16, although of different scope than claim 13, includes features similar to those of claim 13. For example, claim 16 recites, *inter alia*, “means for automatically generating a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.” For at least reasons similar to those presented above in connection with claim 13, claim 16 is distinguishable from *Microsoft*. Accordingly, the § 102(b) rejection of claim 16 based on *Microsoft* should be withdrawn.

Conclusion


Applicant requests the Examiner's reconsideration of the application in view of the foregoing, and the timely allowance of pending claims 1-16.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: October 28, 2005

By: 
Frank A. Italiano
Reg. No. 53,056